Fast Optimistic Network Simulation

Chris Carothers

Paul Belemjian

Rensselaer Polytechnic Institute, Troy, NY

http://www.cs.rpi.edu/~szymansk/sonms.html

email: chrisc@cs.rpi.edu

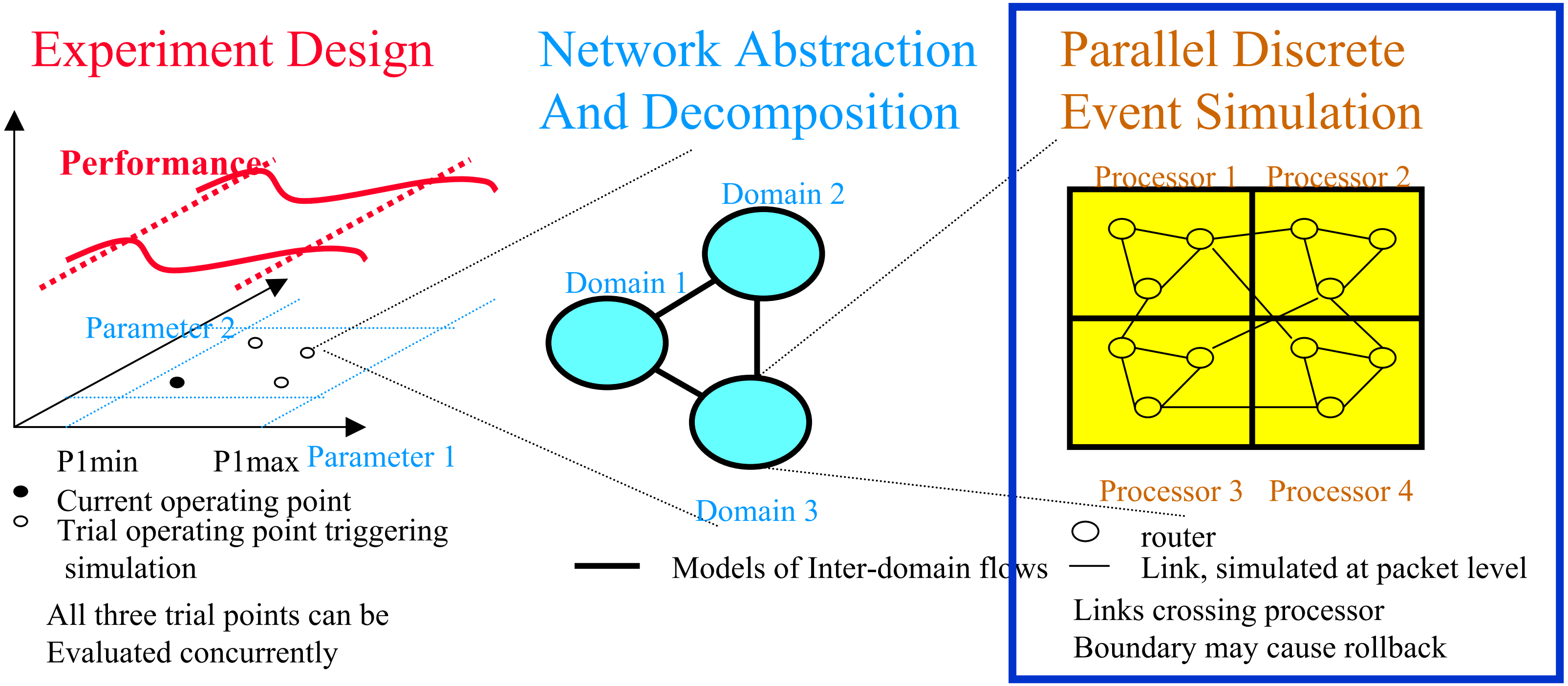
DARPA PI Meeting

April 2, 2001

Novel goals of the research:

- On-Line Network Modeling and Simulation scalable to multiple domains and hundreds of thousands of flows
- Second order traffic and routing control

Topic of this poster



● Current operating point

○ Trial operating point triggering simulation

All three trial points can be Evaluated concurrently

Models of Inter-domain flows

○ router

— Link, simulated at packet level

Links crossing processor Boundary may cause rollback

ROSS: Rensselaer's Optimistic Simulation System

ROSS demonstrates that highly efficient execution is possible when using little *optimistic* memory….

- ***Extreme Performance***
  - ∗ 1,250,000 events/sec, 4 PE case
  - ∗ uses COTS PC hardware
- ***Low Memory***
  - ∗ less than 1% *optimistic* memory for large-scale / low event grain models.
- ***Target Application***
  - ∗ very low event granularity models
  - ∗ wireless / packet-level network models

ROSS' performance is achieved by...

- optimistic synchronization
- Pointer-based, modular implementation framework
- Reverse computation
- Fujimoto's GVT algorithm
- Kernel Processes (KPs)

As a demonstration of ROSS' performance, we have conducted an initial comparison with NS.

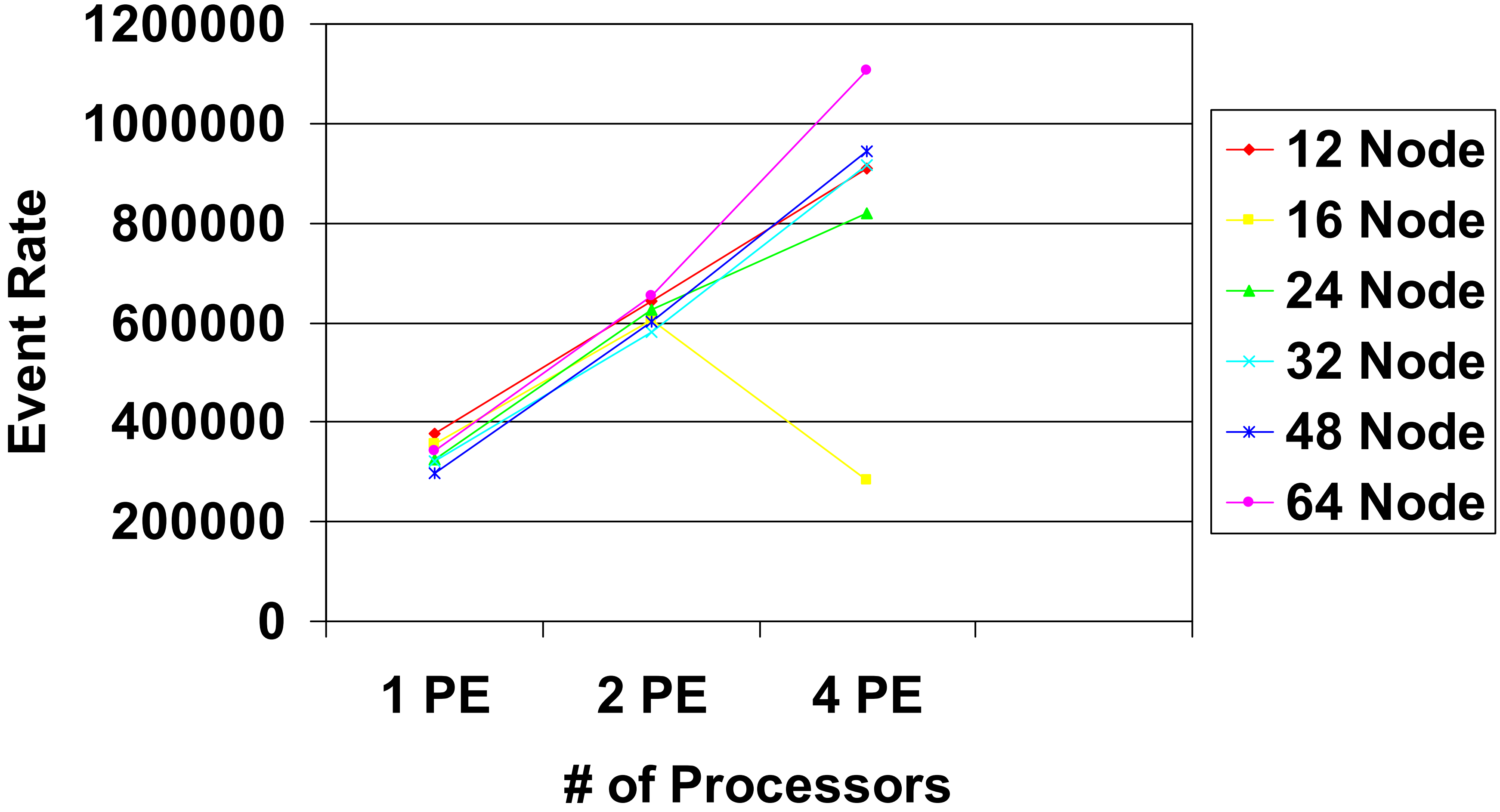
Platform and Experiment Setup

- Network Model
  - TCP/IP
  - uses reverse computation for state saving
  - 3 tier network made up of sources, subnet routers and backbone routers.
  - A network of size N has $N^3 + N^2 + N$ nodes.
- Platform
  - COTS PC-based mulitprocessor systems.
  - Quad processor, Pentuim III
  - 1 GB RAM.
  - Linux OS

TCP/IP Uniprocessor Performance

	Nodes	Speedup	Exec. Time (seconds)	Packets	% Diff
ns	2	-	9.47	1708543	
	8	-	371	31804	
	10	-	1669	30913	
ROSS	2	2.25	4.2	1684238	1.42
	8	256	1.45	26283	17.35
	10	1112	1.5	28325	8.37

ROSS: TCP/IP Parallel Performance



ROSS Embedded Capabilities

- A version of ROSS current runs as an embedded system inside the Linux OS (directly linked).
- User programs invoke ROSS thru the **ross** system call.
- Results and config parameters are pass thru system call.
- New capabilities:
  - * allows for parallel simulations to be embedded into network elemeents.
  - * allows for fine grain control of simulator CPU resources.
  - * allows direct access to OS level network performance statistics
  - * improves simulator performance by 10 to 15% over user-space parallel performance.
  - * potential for improving stability of optimistic synchronization.

Future Work on ROSS/TCP Model...

- ***Strong validation between ns and ROSS/TCP models across a wide rang of configurations***
- ***Implement RED into TCP model.***
- ***Implement PGM model on-top of TCP/IP model***
- ***Flexible specification of network topology.***
- ***Experiment with ROSS as an embedded simulation environment.***